

## RESEARCH ARTICLE

# AUV Dynamic Obstacle Avoidance Method Based on Improved PPO Algorithm

GUOHAO ZHU<sup>1</sup>, ZHOU SHEN<sup>1</sup>, LAIYUAN LIU<sup>1</sup>, SICONG ZHAO<sup>1</sup>, FANGZHENG JI<sup>1</sup>, ZIXIA JU<sup>1</sup>, AND JIALONG SUN<sup>1,2,3,4,5</sup>

<sup>1</sup>School of Geomatics and Marine Information, Jiangsu Ocean University, Lianyungang 222001, China

<sup>2</sup>Jiangsu Marine Resources Development Research Institute, Lianyungang 222005, China

<sup>3</sup>Co-Innovation Center of Jiangsu Marine Bio-Industry Technology, Jiangsu Ocean University, Lianyungang 222001, China

<sup>4</sup>Jiangsu Key Laboratory of Marine Bioresources and Environment/Jiangsu Key Laboratory of Marine Biotechnology, Jiangsu Ocean University, Lianyungang 222001, China

<sup>5</sup>Marine Information Technology Innovation Center of Ministry of Natural Resources, Tianjin 300171, China

Corresponding author: Jialong Sun (h7z3t5ve@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 41774001, in part by the Construction Project of Marine Technology Brand Major in Colleges and Universities of Jiangsu Province, in part by the Priority Academic Program Development Project of Jiangsu Higher Education Institutions (PAPD), in part by the Science and Technology Plan Project of the Lianyungang High Tech Zone under Grant ZD201905, and in part by the Open Fund Project of Marine Information Technology Innovation Center of the Ministry of Natural Resources.

**ABSTRACT** Designing a reasonable obstacle avoidance method for AUV 3D path planning is difficult, and existing obstacle avoidance methods have certain drawbacks. For example, they are only applicable to 2D planar applications and cannot effectively handle dynamic obstacles. To address these problems, we design an obstacle collision prediction model (CPM). Based on the results of the simulation of obstacles' inertial motion, the safety of the AUV navigation is evaluated to improve the model's sensitivity to dynamic obstacles. Then, we enhance the learning ability of the sequence sample data by combining it with a long short-term memory (LSTM) network, thus improving the training efficiency and effect of the algorithm. The trained proximal policy optimization (PPO) network can output reasonable actions in order to control the AUV to avoid obstacles, forming an AUV 3D dynamic obstacle avoidance strategy based on the CPM-LSTM-PPO algorithm. The simulation results show that the proposed algorithm has good generalization in uncertain environments. Moreover, it achieves dynamic AUV obstacle avoidance in different three-dimensional unknown environments, providing theoretical and technical support for real path planning.


**INDEX TERMS** AUV, dynamic obstacle avoidance, deep reinforcement learning, proximal policy optimization algorithm, collision prediction model.

## I. INTRODUCTION

The autonomous underwater vehicle (AUV), which serves as a lightweight underwater observation tool, has gradually become an important piece of equipment for countries to explore marine resources and strengthen the national navy. Among other advantages, AUVs are small, easy to control, and highly intelligent. [1], [2] In a complex and changing marine environment, AUVs' safety obstacle avoidance technology not only supports their navigational and operational

functions but is also an important part of their navigation control technology. As various countries have expanded their efforts in ocean exploration, further improvement of AUVs' dynamic obstacle avoidance capability in complex marine environments has become a key avenue for increasing their effectiveness [3], [4].

In AUVs' application scenario, the complex and dense dynamic obstacles of an uncertain environment pose a huge challenge to safe navigation. Traditional obstacle avoidance methods (such as the A\* algorithm, artificial potential field method, Voronoi diagram, RRT algorithm, swarm intelligence algorithm [5], [6], [7], [8], [9], etc.) have been used

The associate editor coordinating the review of this manuscript and approving it for publication was Usama Mir .

to avoid obstacles when the environmental information is known. However, due to the dense dynamic obstacles in uncertain environments, AUVs cannot obtain the motion information of dynamic obstacles in advance. Therefore, traditional methods cannot be effectively applied to real-time obstacle avoidance. In addition, the complexity and variability of the uncertain environment imposes higher requirements on the speed of the obstacle avoidance algorithm. Traditional methods are overly reliant on environmental dynamic models and AUV models; as a result, the accuracy of the models greatly affects these methods' performance. On the one hand, simple models cannot adequately characterize the complexity of the environment. On the other, complex models are too computationally intensive, which not only wastes computational resources but also takes too long to meet the needs of AUVs in uncertain environments. Therefore, it is necessary to design a method that can realize dynamic obstacle avoidance for AUVs in an uncertain environment [10], [11].

With the development of artificial intelligence, more and more advanced intelligent algorithms have been applied in various fields to solve problems that cannot be solved by traditional algorithms. Among intelligent decision-making algorithms, deep reinforcement learning methods stand out for their powerful high-dimensional information perception, understanding, and nonlinear processing capabilities. Wu Yahui et al. proposed a model of obstacle avoidance built on a modified artificial potential field method with the obstacle information in the environment and the posture and angle information of the robot movement, thus achieving autonomous movement of the robot in unfamiliar scenes [12]. Xiong Juntao et al. addressed the problem that the range repulsion of the artificial potential field method affected the shortest path planning. They proposed a method for setting the directional penalty obstacle avoidance function, which converted the obstacle range penalty into a single direction penalty. By establishing a virtual robot motion collision model, the direction penalties were given selectively by the analysis results of the model [13]. Liu Qingjie et al. note that rewarding sparseness in the learning process makes it difficult to obtain better results; therefore, to improve the reward mechanism, they increase real-time rewards and punishments as a supplement to solve the problem of long learning time and unstable training [14]. Sun Lixiang et al. developed a reward function for reinforcement learning based on human spatial behavior. In this approach, states in which the robot angle changes significantly are punished to achieve the requirements of comfortable obstacle avoidance [15]. Mirowski et al. used deep reinforcement learning to make navigation decisions in a grid environment. However, they performed the task in a static obstacle environment, thus failing to verify the algorithm's practicability in an uncertain environment [16]. Qiao et al. used CMAC (cerebellar model arithmetic computer) and SARSA (a temporal-difference reinforcement learning method) to complete automatic obstacle avoidance in unknown environments. However, this method is limited to collision avoidance

for a single obstacle [17]. Finally, Zhou Bin et al. used the received signal strength to define the reward value and employed Q-learning to complete AUV path planning and obstacle avoidance. Nevertheless, as the application scenario is simple, they also fail to consider uncertain environments with a large number of dynamic obstacles [18].

In summary, although the above methods have achieved good results in their respective environments, there are still some shortcomings, mainly in the following two areas. First, most algorithms only perform obstacle avoidance or path planning in a static environment and cannot deal with dynamic obstacles; as a result, they are difficult to apply in uncertain environments. Second, due to the environment type for obstacle avoidance and the consideration of model complexity and computational intensity, deep reinforcement learning algorithms can only be applied to the field of two-dimensional plane obstacle avoidance, which is far from a three-dimensional environment. Therefore, these algorithms have certain limitations in guiding real-world applications.

Aiming at the above problems and relying on existing research, this study proposes an AUV dynamic obstacle avoidance method based on proximal policy optimization (PPO) for uncertain environments with dense dynamic obstacles. The specific steps of the study are as follows. First, the obstacle collision prediction model is designed. Based on the results of the simulation of the obstacles' inertial motion, the safety of the AUV navigation is evaluated to improve the model's sensitivity to dynamic obstacles. The introduction of the long short-term memory network transforms the environmental state into a high-dimensional perception situation, strengthening the network's ability to learn time-series obstacle avoidance data. Thus, we propose an AUV dynamic obstacle avoidance method based on a CPM-LSTM-PPO algorithm, which makes full use of the plasticity of the offline training of the neural network and real-time online use. Finally, we design various obstacle avoidance simulation experiments to compare the proposed method with other algorithms in order to verify its effectiveness and superiority.

## II. PPO ALGORITHM AND LSTM NETWORK

### A. PROXIMAL POLICY OPTIMIZATION ALGORITHM

The proximal policy optimization algorithm [19] is an improved deep reinforcement learning algorithm proposed by OpenAI in 2017. In the same year, DeepMind showed that the agent could explore complex skills without special instructions by training a PPO. This further proved that the PPO algorithm can be better applied to the tasks of continuous control and continuous plotting.

PPO is a new type of policy gradient (PG) algorithm. The main philosophy of the PG algorithm is to use gradient boosting to update the policy  $\pi$  in order to maximize the expected reward. In the PG algorithm, the objective function of the network parameter  $\theta$  update is as follows:

$$L^{PG}(\theta) = E_t [\lg \pi_{\theta}(a_t | s_t) \times A_t] \quad (1)$$

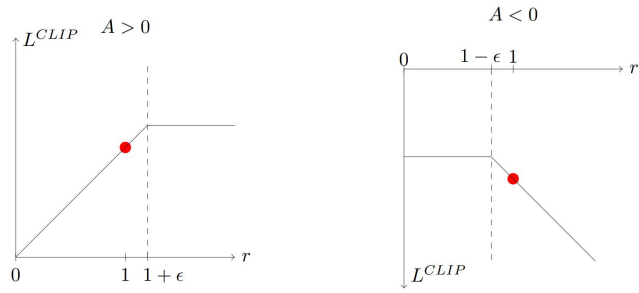


FIGURE 1. Schematic diagram of the clip function of PPO.

The biggest advantage of the PG algorithm is that it can choose actions in a continuous space. Its disadvantage is that although it is sensitive to step size, choosing a suitable step size is difficult. To address this shortcoming, this paper first uses the ratio of the action probability  $\pi_\theta(a|s)$  under the current strategy to the action probability  $\pi_{\theta_{old}}(a|s)$  of the previous strategy to observe the effect of the agent’s action. The ratio of old and new strategies is recorded as

$$r_t(\theta) = \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_{old}}(a_t | s_t)} \quad (2)$$

If the reward function  $r_t(\theta) > 1$ , it indicates that the probability of the action occurring under this policy is higher than that of the previous policy; if  $0 < r_t(\theta) < 1$ , the probability is lower than the previous policy. The objective function can be designed as follows:

$$L^{CPI}(\theta) = E_t \left[ \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_{old}}(a_t | s_t)} A_t \right] = E_t [r_t(\theta) A_t] \quad (3)$$

Second, to avoid policy mutation during the process of parameter updating, the objective function (Formula (3)) must be constrained. The PPO algorithm improves the stability of training agent behavior by constraining policy updates to a small range. There are two kinds of constraints that the PPO algorithm can adopt: limiting the KL divergence or truncation. In practical applications, researchers have found that the truncated method works better. Therefore, the objective function of PPO is optimized as follows:

$$L^{CLIP}(\theta) = E_t [\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)A_t)] \quad (4)$$

Among them,  $\epsilon$  is a truncation constant used to assist in setting the range of policy updates; it is usually set to 0.1 or 0.2. The clip function is a truncation function that limits the value of the old and new policy parameters  $r_t(\theta)$  to the interval  $[1 - \epsilon, 1 + \epsilon]$ , as shown in Figure 1. The objective function uses the min function to represent the smaller value between the probability ratio of the old and new strategies and the truncation function.

When the advantage function  $A$  is positive, it means that the current action has a positive effect on the optimization goal. Therefore, the probability of its occurrence should

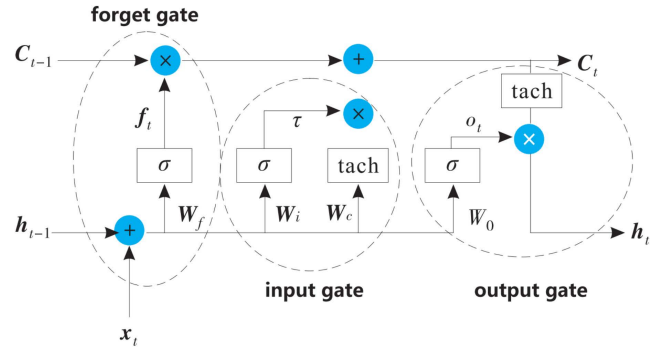


FIGURE 2. LSTM cell structure.

be increased, but the update range should be limited to below  $1 + \epsilon$ . When  $A$  is negative (indicating that the current behavior is negative), it should be blocked while reducing its probability to  $1 - \epsilon$ .

The core philosophy of the PPO algorithm is to avoid the use of large policy updates in order to solve the problem of difficult step size determination and low data efficiency in the PG algorithm. This greatly reduces the difficulty of debugging by researchers.

### B. LONG SHORT-TERM MEMORY NETWORK

Each unit of the LSTM network can be divided into a forget gate  $f_t$ , input gate  $i_t$ , and output gate  $o_t$  (Fig. 2).

Among them, the forget gate uses the sigmoid function to determine whether the output  $h_{t-1}$  and cell state  $C_{t-1}$  of the network at the previous time continue to exist in the cell state  $C_t$  of the current network. The calculation formula of the forget gate is as follows:

$$f_t = \sigma(W_f \cdot g[h_{t-1}, x_t] + b_f) \quad (5)$$

In formula (5),  $W_f$  is the weight matrix;  $b_f$  is the offset;  $x_t$  is the input of the current network; and  $g$  represents vector splicing.

The input gate multiplies the information output by the sigmoid function and the tach function to determine how much of the current input  $x_t$  is to be transferred to the cell state  $C_t$ . The formula of the input gate is as follows:

$$i_t = \sigma(W_i \cdot g[h_{t-1}, x_t] + b_i) \text{tach}(W_c \cdot g[h_{t-1}, x_t] + b_c) \quad (6)$$

The output gate also uses the information output by the sigmoid function and the tach function to determine how much of the unit state  $C_t$  can be transferred to the current output  $h_t$ . The formula of the output gate is as follows:

$$h_t = \sigma(W_0 \cdot g[h_{t-1}, x_t] + b_0) \cdot \text{tach}(C_t) \quad (7)$$

### III. CPM-LSTM-PPO ALGORITHM

The main goal of this paper is to allow the AUV to find a reasonable path to the target position within the specified step size. Here, “reasonable” refers to distinguishing obstacles in different directions and speeds so that the AUV behavior

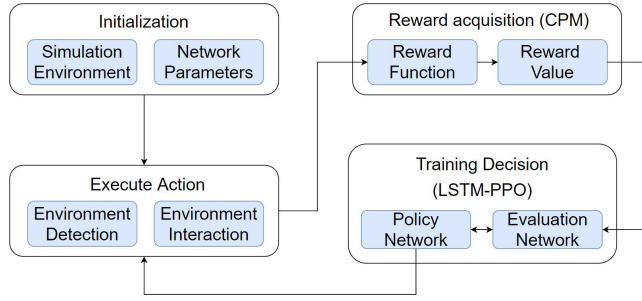


FIGURE 3. Schematic diagram of AUV training process.

path is closer to reality. In this paper, the PPO algorithm is used for 3D dynamic obstacle avoidance tasks in an unknown environment with multiple obstacles. The training process consists of four stages: initialization, action execution, reward acquisition, and training decision-making (Fig. 3).

First, a reasonable environment state and action state are designed. Second, AUV uses sonar to detect environmental information and collect data. Then, it inputs these data as feature vectors combined with a reward function into a neural network for training. Finally, the optimal action is selected according to the exploration strategy, and the output reaches the next visual observation. AUV continuously loops and iterates the three stages—executing actions, obtaining rewards, and making training decisions—until the training is completed.

### A. COLLISION PREDICTION MODEL

The first step is to construct a 3D coordinate system. Take the position of the AUV when the active sailing function is turned on as the origin  $(0, 0, 0)$ . The heading is the positive direction of the  $y$ -axis. The positive direction of the  $x$ -axis is in the horizontal direction, perpendicular to the heading direction and pointing to the right. The positive  $z$ -axis direction is perpendicular to the heading direction and pointing to the water surface. The second step is to map the detected obstacle recognition frame to the map and update the coordinate information of obstacles and the AUV in real time.

Assuming that the velocity  $v_{obs}$ , pitch angle  $\theta_{obs}$ , and yaw angle  $\psi_{obs}$  of the obstacle within  $t$  seconds are all fixed, the position of the coordinate system in the previous frame of the obstacle measured by the sonar is  $(x_1, y_1, z_1)$ , and the current frame position of the obstacle is  $(x_{obs}, y_{obs}, z_{obs})$ , the speed of obstacle navigation is as follows:

$$v_{obs} = \sqrt{(x_1 - x_{obs})^2 + (y_1 - y_{obs})^2 + (z_1 - z_{obs})^2} / t \quad (8)$$

The yaw angle is

$$\psi_{obs} = \arctan((y_1 - y_{obs}) / (x_1 - x_{obs})) \quad (9)$$

The pitch angle is

$$\theta_{obs} = \arctan\left(\frac{(z_1 - z_{obs}) / \sqrt{(x_1 - x_{obs})^2 + (y_1 - y_{obs})^2}}{1}\right) \quad (10)$$

These formulas allow the dynamic information of the obstacle to be judged.

After storing the above information, a three-dimensional map of the absolute coordinates of obstacles, target positions, and the AUV itself is formed.

To build a collision prediction model, the collision distance must be calculated first.

Assuming that the position of the current frame of the AUV is  $(x_{auv}, y_{auv}, z_{auv})$ , the movement of the coordinates after completing a step navigation action is  $(\Delta x_{auv}, \Delta y_{auv}, \Delta z_{auv})$ . That is, the position of the AUV after completing a step navigation action is  $(x_{auv} + \Delta x_{auv}, y_{auv} + \Delta y_{auv}, z_{auv} + \Delta z_{auv})$ , and the time required for the AUV to complete a step sailing action is  $\Delta t$  seconds ( $\Delta t$  is on the order of milliseconds).

The amount of movement of the obstacle in the  $x$ -axis after  $\Delta t$  seconds is  $\Delta x_{obs} = v_{obs} \Delta t \cos \theta_{obs} \cos \psi_{obs}$ .

The amount of movement on the  $y$ -axis is  $\Delta y_{obs} = v_{obs} \Delta t \cos \theta_{obs} \sin \psi_{obs}$ .

The amount of movement on the  $z$ -axis is  $\Delta z_{obs} = v_{obs} \Delta t \sin \theta_{obs}$ .

That is, the coordinate of the obstacle after  $\Delta t$  seconds is  $(x_{obs} + \Delta x_{obs}, y_{obs} + \Delta y_{obs}, z_{obs} + \Delta z_{obs})$ .

Then, after  $\Delta t$  seconds, the distance between the AUV and the obstacle in (11), as shown at the bottom of the next page.

The obstacle distance is scored according to  $dist$ , from which the obstacle distance reward  $R_{\Delta t}$  is obtained. In this paper, the safe distance is set to 5 meters, the general distance is 3.5 meters, and the dangerous distance is 2 meters. Therefore, the AUV obstacle distance reward  $R_{\Delta t}$  is as follows:

$$R_{\Delta t} = \begin{cases} 2, & dist > 5m \\ 1, & 3.5m < dist \leq 5m \\ -1, & 2m < dist \leq 3.5m \\ -2, & dist \leq 2m \end{cases} \quad (12)$$

AUV dynamic obstacle avoidance is a continuous process, and the navigation action taken in the current step will greatly affect the next action. Therefore, focusing exclusively on the effect of the current action will often affect overall obstacle avoidance. At the same time, considering the inertia of object motion, both the AUV and dynamic obstacles are unlikely to change their original speed and heading within a few tens of  $\Delta t$  seconds. Therefore, it may be assumed that the AUV takes the current navigation action over the course of the next several dozen steps, and the influence of inertial motion is estimated to calculate the overall AUV obstacle distance reward  $G_{m\Delta t}$ :

$$\begin{aligned} G_{m\Delta t} &= R_{\Delta t} + \gamma R_{2\Delta t} + \gamma^2 R_{3\Delta t} + \dots = \sum_{k=1}^m \gamma^{k-1} R_{k\Delta t} \\ G_{m\Delta t} &= R_{\Delta t} + \gamma R_{2\Delta t} + \gamma^2 R_{3\Delta t} + \gamma^3 R_{4\Delta t} + \dots \\ &= R_{\Delta t} + \sum_{k=1}^m \gamma^k R_{(k+1)\Delta t} \end{aligned} \quad (13)$$

In formula (13),  $G_{m\Delta t}$  is the total obstacle distance reward obtained in  $m$  steps.  $R_{i\Delta t}$  is the obstacle distance reward at

the  $n$ th step (that is, after  $n \Delta t$  seconds).  $\gamma$  is the attenuation factor, which is between (0, 1) because the closer  $R_{\Delta t}$  is, the greater its impact on the algorithm. As  $A$  gradually becomes farther, the accuracy gradually decreases due to its predictability. The addition of  $\gamma$  prevents the collision prediction model from being either too short-sighted or too long-term.

Considering the computational performance of AUV, after simulation experiments, we obtain  $m = 30$ ,  $\gamma = 0.95$ :

$$G_{30\Delta t} = \sum_{k=1}^{30} 0.95^{k-1} R_{k\Delta t} \quad (14)$$

The collision prediction model in this paper is divided into four levels: A (safe), B (lower collision risk), C (higher collision risk), and D (extremely dangerous). We substitute  $G_{30\Delta t}$  into the following formula to obtain the AUV's estimated collision rating  $S_q$  for this obstacle:

$$\begin{cases} A, & G_{30\Delta t} > 25.13 \\ B, & 18.85 < G_{30\Delta t} \leq 25.13 \\ C, & 6.28 < G_{30\Delta t} \leq 18.85 \\ D, & G_{30\Delta t} \leq 6.28 \end{cases} \quad (15)$$

Assuming that  $q$  obstacles are identified on the same frame of the sonar image, we repeat the above steps for these  $q$  obstacles to obtain collision prediction set  $S$ :

$$S = \{S_1, S_2, S_3, \dots, S_q\} \quad (16)$$

## B. STATE SPACE AND ACTION SPACE

The environment model of the AUV must consider the target position, boundary information, and obstacle collision prediction model to engage in reasonable behavior to avoid a collision. A variety of obstacles are set up in this paper's simulation environment and change randomly within a certain range. Due to the huge number of states and actions in the continuous high-dimensional space, it is difficult for the algorithm to converge. Therefore, we discretize information such as obstacles around the AUV into a finite number of states and formulate the state space reasonably. The state space is defined as follows:

$$s_t = (x_{auv}, y_{auv}, z_{auv}, dist_{end}, step, S) \quad (17)$$

In formula (17),  $(x_{auv}, y_{auv}, z_{auv})$  is the position of the AUV's current frame, and  $dist_{end}$  is the distance between the AUV and the target position. In addition,  $step$  is the number of steps taken to navigate, and  $S$  is the collision prediction set.

To speed up the convergence of the network model, the action space consists of six discrete actions:

$$a_t = (a_0, a_1, a_2, \dots, a_5) \quad (18)$$

Among them,  $a_0, a_1, a_2, a_3, a_4$ , and  $a_5$  are 0.2 m forward in the directions of  $+x$ -axis,  $-x$ -axis,  $+y$ -axis,  $-y$ -axis,  $+z$ -axis, and  $-z$ -axis, respectively.  $+$  and  $-$  indicate the forward and reverse directions, respectively.

## C. DESIGN OF CPM-LSTM-PPO ALGORITHM FRAMEWORK

Since the underwater environment has highly dynamic, high-dimensional characteristics and complexity, simply using the fully connected neural network in the PPO algorithm to approximate the policy function and evaluation function is inadequate. The policy network and evaluation network in this paper use the LSTM network framework. First, the LSTM network is introduced to extract features from a high-dimensional environmental situation, output useful perception information, and enhance the learning ability of serial sample data. Then, it approximates the policy function and evaluation function through a fully connected neural network. Figure 4 shows the framework of the CPM-LSTM-PPO algorithm.

For the policy network part, six nodes are set up in the input layer, corresponding to the six states of  $s_t$ . The hidden layer sets up the LSTM layer and the fully connected layer. The LSTM layer sets up three network units, and the fully connected layer is designed as three layers, all of which use  $\tanh$  as the activation function. The output layer sets a node and uses softmax as the activation function for the simplified discrete action  $a_t$ . Figure 5 shows the policy network framework.

## D. REWARD AND PUNISHMENT FUNCTION

In deep reinforcement learning algorithms, all objectives can be described by maximizing the expected cumulative reward. Therefore, AUVs can learn the correct strategy from feedback signals when interacting with the environment.

The reward and punishment function is the key to determining whether the deep reinforcement learning network model can successfully converge. In this paper, the reward and punishment function  $R$  is mainly composed of three parts: the reward and punishment for distance change  $R_1$ , the reward and punishment for collision prediction  $R_2$ , and the reward and punishment for arrival, out of bounds, and collision occurrence  $R_3$ .

$R_1$  means that if the AUV is closer to the target position after performing a step action, it will give an appropriate reward; otherwise, it will give a penalty.  $R_2$  indicates that the collision prediction reward and punishment are given according to each rating  $S_q$  in  $S$ .  $R_3$  means that the AUV will give a completion reward when it reaches the target position and a failure penalty if the coordinates exceed the delimited boundary or collide.

$$dist = \sqrt{(x_{auv} + \Delta x_{auv} - x_{obs} - \Delta x_{obs})^2 + (y_{auv} + \Delta y_{auv} - y_{obs} - \Delta y_{obs})^2 + (z_{auv} + \Delta z_{auv} - z_{obs} - \Delta z_{obs})^2} \quad (11)$$

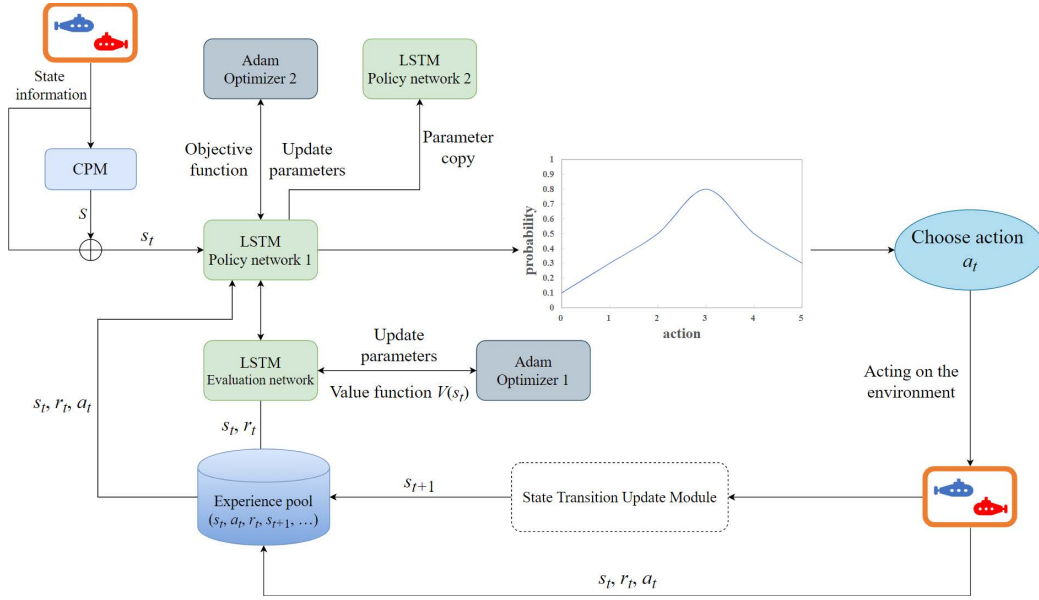


FIGURE 4. CPM-LSTM-PPO algorithm framework.

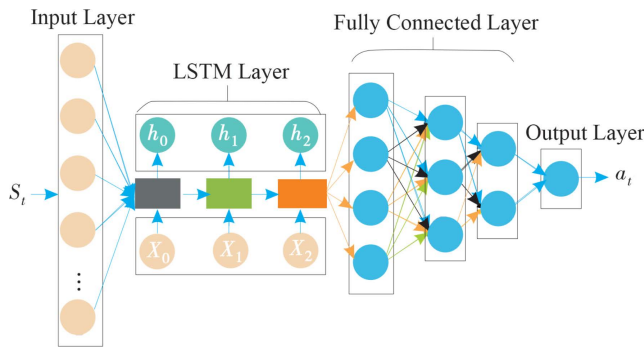


FIGURE 5. Policy network framework.

The reward and punishment function is designed as follows:

$$R = R_1 + R_2 + R_3 \quad (19)$$

$$R_1 = \begin{cases} 3, & dist_{end} < predist_{end} \\ -4, & dist_{end} \geq predist_{end} \end{cases} \quad (20)$$

$$R_2 = \begin{cases} 0.2, & S_q = A \\ 0.1, & S_q = B \\ -10, & S_q = C \\ -30, & S_q = D \end{cases} \quad (21)$$

$$R_3 = \begin{cases} 30000, & dist_{end} < 0.1 \\ -30000, & (out\ of\ bounds\ or\ collision) \end{cases} \quad (22)$$

In formula (20),  $predist_{end}$  represents the distance between the AUV before performing the action and the target position.

Appropriate safety rewards and punishments and severe dangerous action penalties through collision prediction allow the algorithm to take safe obstacle avoidance actions.

To prevent situations in which the AUV can never reach the target position, this paper sets a maximum limit number of steps  $\sigma$  of the map. This value changes according to map size:

$$\sigma = \lambda * (l * w * h) \quad (23)$$

where  $l, w, h$  are the length, width, and height of the map, respectively.  $\lambda$  is a parameter related to the complexity of the map; a larger value should be set for a more complex map.

$R \geq 30,000$ , or  $R \leq -10,000$ , or  $step\ number \geq \sigma$ , will immediately end the current round of episodes.

#### IV. EXPERIMENTS

This paper uses the Python-based physics engine PyBullet to build the simulation environment. The computer configuration for AUV training is as follows: the hardware environment is an Intel i5-7300HQ processor, 16GB memory, and NVIDIA GeForce GTX 1050Ti graphics card. The software environment is Python 3.10.

In this paper, several experiments are designed to verify the algorithm's effectiveness.

Experiment 1 is an AUV dynamic obstacle avoidance simulation experiment based on the CPM-LSTM-PPO algorithm.

Experiment 2 is a comparison experiment between the algorithm in this paper and other algorithms.

Experiment 3 examines a random dynamic obstacle avoidance scene.

##### A. SIMULATION EXPERIMENT

###### 1) ENVIRONMENT MODEL AND TRAINING PARAMETERS

The basic training simulation environment designed in this paper has certain representativeness (Fig. 6). The length, width, and height of the training environment are 55 m, 18 m,

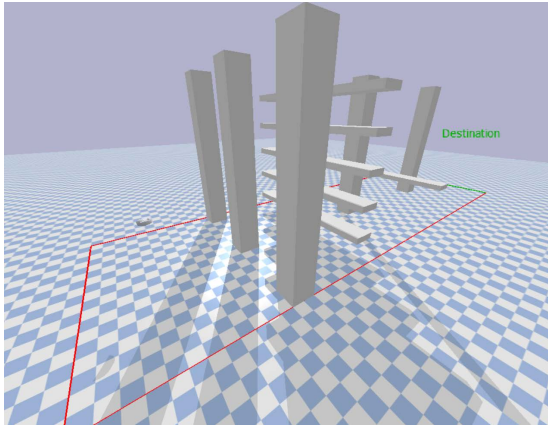


FIGURE 6. AUV basic training environment.

TABLE 1. PPO algorithm parameters.

Parameters	Value
Learning rate	3e-4
Batch_size	64
N_steps	2,048
Clip_range	0.2
Gamma	0.99
Gae_lambda	0.95

and 14 m, respectively. The red line is the boundary line, the green line is the target position, and the orange line is a segment of the navigation trajectory generated by the AUV every 40 steps. The AUV first traverses three uprights and then five lateral static obstacles. Then, it passes through two dynamic obstacles that move left and right and one that moves up and down. The obstacles follow uniform round-trip linear motion.

In the experiment, the continuous observation vector space is used, and the eigenvectors are applied to represent the observation results of the intelligent agent at each step. Each AUV continuously learns and explores according to the method proposed in this paper. During the training process,  $R$  is always followed for certain rewards and punishments. The entire scene is reset at the end of each round or if the AUV goes out of bounds. The total number of training iterations in this experiment is 6,000, and the PPO parameter settings are shown in Table 1.

## 2) EXPERIMENTAL RESULTS AND ANALYSIS

The average reward obtained every 10 rounds during the training process, as well as the number of steps taken by the AUV to reach the target position each time, was recorded (Figs. 7 and 8). As the number of iteration rounds increases, when the algorithm iterates to about 2,000 rounds, the average reward has increased from a negative value to 0. This indicates that the CPM-LSTM-PPO algorithm has gained

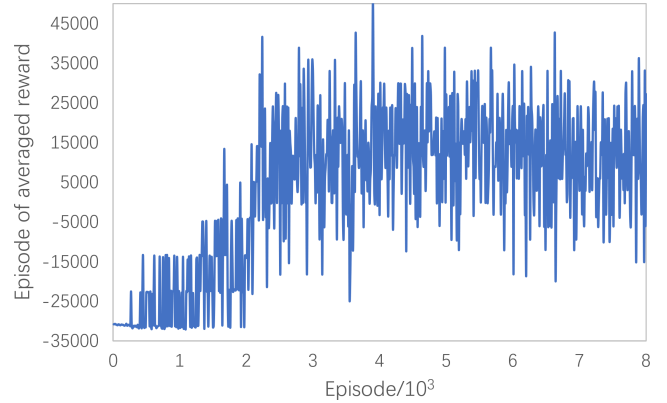


FIGURE 7. Average reward.

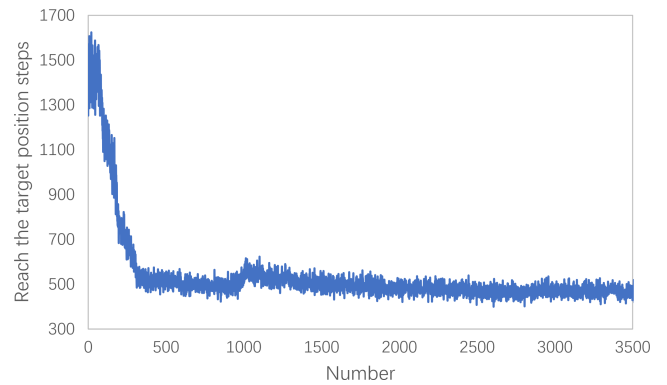


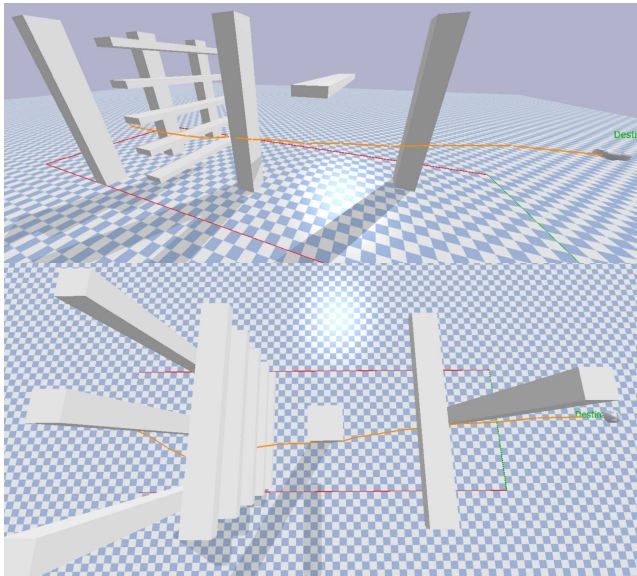
FIGURE 8. Number of steps used.

some obstacle avoidance experience. When the algorithm iterates to the 3,000th round, the average reward for every 10 rounds fluctuates around 15,000. The reason why the average reward fails to converge above 30,000 rounds is that the failed attempts will lower the average reward every 10 rounds; as a result, the algorithm's success rate is below 100%. Figure 8 shows that after the AUV first reaches the target position, the number of steps used gradually decreases. After reaching the target position 1,500 times, the number of steps used tends to be stable and continues to fluctuate around 500 steps, indicating that the CPM-LSTM-PPO algorithm tends to converge.

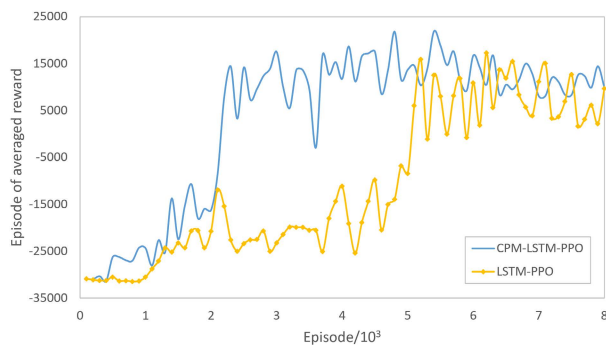
Figure 9 shows the path planned by the training model using the CPM-LSTM-PPO algorithm. The AUV selects the safest position in the middle when crossing the static and horizontal columns, indicating that the model has high path smoothness and has learned the target position and dynamic obstacle avoidance function.

In the same experimental environment, this paper divides the reward and punishment functions into two cases for comparison: first, a complete reward and punishment mechanism, namely  $R = R_1 + R_2 + R_3$ ; and second, a version without the collision prediction model, that is,  $R = R_1 + R_3$ .

Figure 10 compares the average rewards per hundred rounds based on different reward and punishment functions.



**FIGURE 9.** The planning path diagram of the proposed algorithm training model.



**FIGURE 10.** Comparison of training situations based on different reward functions.

The blue line represents the first training case (the CPM-LSTM-PPO algorithm), and the orange line represents the second case. It can be intuitively seen from the figure that the blue line achieves a better cumulative reward value in fewer iterations; the average reward has reached 15,000 after 3,000 rounds of training. In the case without the collision prediction model, the AUV has a longer learning time in the early training, and the average reward reaches 10,000 when training 5,000 times. The experimental results show that adding a collision prediction model can improve AUV training efficiency and speed up the AUV's exploration of the environment.

## B. COMPARATIVE EXPERIMENT

For more complex multi-dynamic obstacle scenarios, this paper performs AUV dynamic obstacle avoidance tasks based on the DQN algorithm, the TRPO algorithm, the LSTM-PPO algorithm, and the proposed CPM-LSTM-PPO algorithm. In particular, we compare the average reward obtained in the

same scenario and the number of steps taken to reach the target position.

The multi-dynamic obstacle scene consists of seven cubes that engage in round-trip linear motion with different headings and speeds. Figure 11 shows the average rewards per hundred rounds obtained by the four algorithms in a multi-dynamic obstacle environment. The CPM-LSTM-PPO algorithm has less fluctuation in the early training process than the DQN and TRPO algorithms. All three algorithms start to converge around 6,000 rounds, but the LSTM-PPO algorithm gradually decreases after achieving a high reward score in 2,000 rounds. The algorithm in this paper uses the memory function of the LSTM neural network to accumulate higher rewards with the help of the collision prediction model. In the later stages of training, the average reward convergence per ten rounds fluctuates around 22,000. The DQN, TRPO, and LSTM-PPO algorithms converge at 5,000, 8,000, and 10,000 rounds, respectively. These results indicate that the CPM-LSTM-PPO algorithm model has high performance, strong stability, and better generalization ability.

Figure 12 shows the algorithm's obstacle avoidance process in a multi-dynamic obstacle scene. It can be clearly seen that the AUV maneuvers to avoid cubic obstacles, always maintains a safe distance from the obstacles, and completes the obstacle avoidance task in the process of driving to the target position. The path is smooth, without sharp turns, and without many redundant sections. The track of the LSTM-PPO algorithm without collision prediction is similar to this, but it does not keep enough distance from the obstacles.

Figure 13 is the obstacle avoidance diagram of the comparison algorithm in the multi-dynamic obstacle scene. The planned paths of the DQN and TRPO algorithms are the same. They all make only the necessary evasive maneuvers, and the path tends to be a smooth arc, which results in the fewest steps used and the shortest path. However, the downside is that they ignore the need to keep a safe distance from the obstacles, giving the system a lower score. This is also one of the factors explaining why the final average reward of the DQN, TRPO, and LSTM-PPO algorithms is lower than that of the proposed algorithm in this paper.

Figure 14 provides a comparison chart of the number of steps used in a multi-dynamic obstacle scene. The number of steps used by the CPM-LSTM-PPO algorithm decreases rapidly after reaching the target position 200 times, while the number of steps converges around 570 steps after 300 times. Meanwhile, the number of steps used by the LSTM-PPO algorithm gradually decreases with the increase in the number of successes, finally converging around 450 steps after 400 times. Both the DQN and TRPO algorithms converge to around 260 steps after reaching the target position 600 times. Table 2 shows the obstacle avoidance results of each algorithm in a multi-dynamic obstacle scene after 5,000 rounds of training. Although the CPM-LSTM-PPO algorithm uses the most average steps, its 70.76% success rate is much higher than the 56.66% of the DQN algorithm and 52.06%



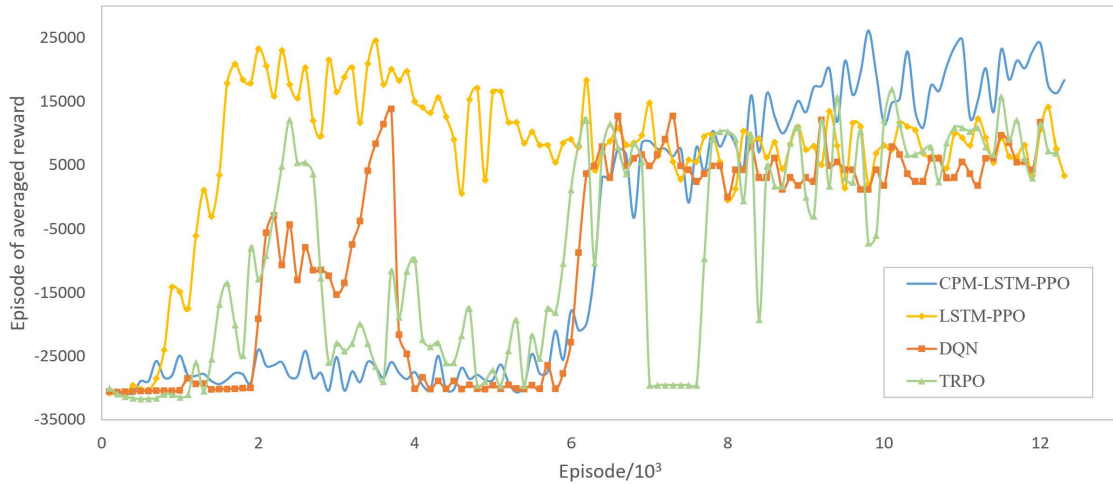


FIGURE 11. Comparison of average rewards in a multi-dynamic obstacle scene.

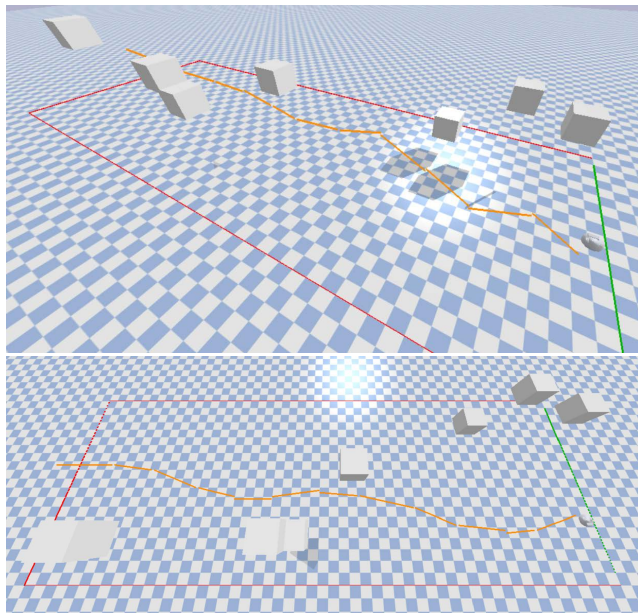
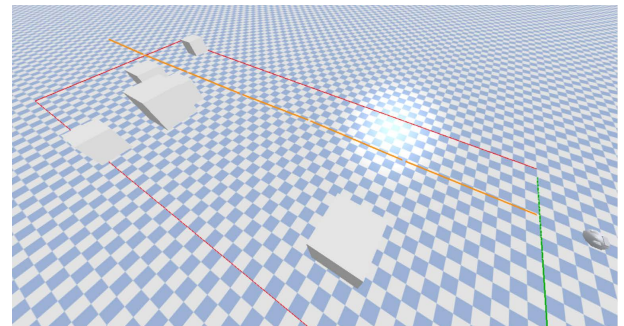
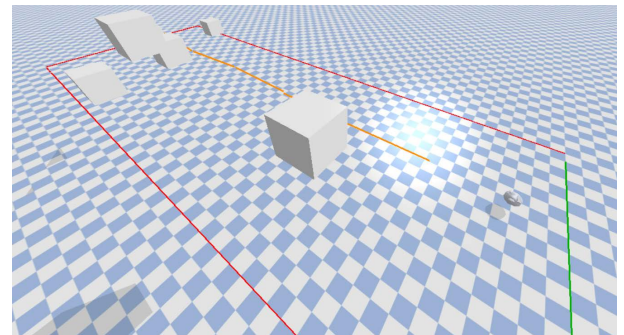


FIGURE 12. Diagram of the obstacle avoidance process of the algorithm in this paper in a multi-dynamic obstacle scene.



(a) DQN algorithm



(b) TRPO algorithm

FIGURE 13. Obstacle avoidance diagram of the comparison algorithm in a multi-dynamic obstacle scene.

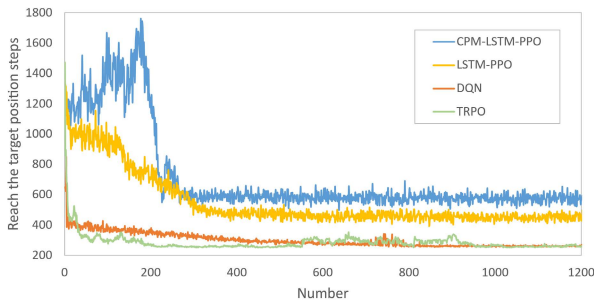
of the TRPO algorithm. The CPM-LSTM-PPO algorithm takes more evasive actions to maintain a safe distance from obstacles, using the collision prediction model to improve the AUV’s sensitivity to dynamic obstacles. It thus achieves a higher obstacle avoidance success rate.

## V. DISCUSSION

Although the obstacles are dynamic in the above two experimental scenarios, their initial position, heading, and speed are all fixed. To give the algorithm good generalizability, it is necessary to randomize the position and motion information of each obstacle in the training environment. This will inevitably lead to longer algorithm training times and will require better obstacle avoidance performance from the algorithm.

To test the obstacle avoidance effect of the CPM-LSTM-PPO algorithm in a random environment, we modify the multi-dynamic obstacle scene used in this paper. The initial positions  $(x, y, z)$  of the seven cube obstacles will appear randomly among  $([-8, 8], [0, 45], [2, 10])$ . The speed of each step is random between  $[0.02, 0.15]$ , and the heading is uncertain. The obstacles engage in back-and-forth motion after touching the boundary.

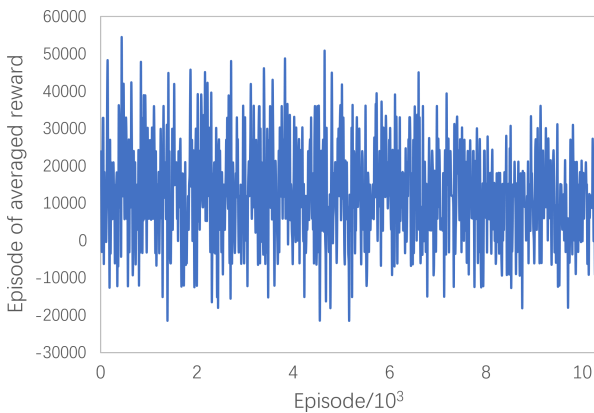
Transfer learning makes the training of the target task more flexible, efficient, and realistic by applying the experience learned from the source task to the target task.



**FIGURE 14.** Comparison of steps used in multi-dynamic obstacle scenarios.

**TABLE 2.** Obstacle avoidance results in multi-dynamic obstacle scenarios.

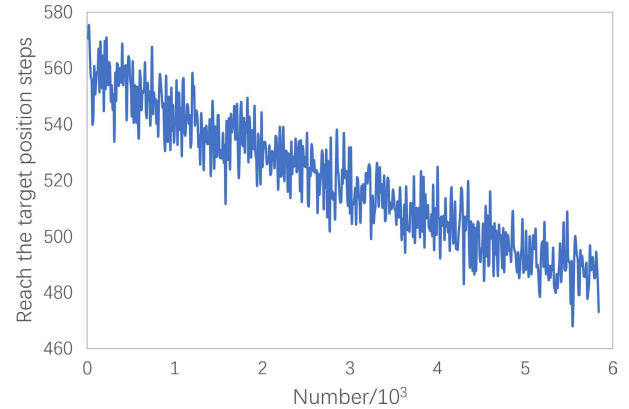
Algorithm	Success	Fail	Average steps	Success rate
DQN	2,833	2,167	261.71	56.66%
TRPO	2,603	2,397	256.15	52.06%
LSTM-PPO	3,157	1,843	451.91	63.14%
CPM-LSTM-PPO	3,538	1,462	567.29	70.76%



**FIGURE 15.** Average reward.

The implementation methods of transfer learning include instance-based, feature-based, model-based, and relation-based methods. In this paper, model-based transfer learning is used to initialize the weights of the CPM-LSTM-PPO model network by using the model parameter pretrained in the multi-dynamic obstacle scenario, replacing the original random initialization operation, and completing global fine-tuning. The rest of the training process is carried out as usual. This can achieve a faster model fit and improve the results.

Figures 15 and 16 show the average reward per 10 rounds and the average number of steps taken by the AUV to reach the target position per 10 times, respectively. With the prior knowledge of transfer learning, the model achieves high scores at the beginning of training, and the average reward fluctuates around 15,000. When the algorithm iterates to the 10,000th round, the average reward fluctuation decreases, but the score decreases as well. Figure 16 shows that the average number of steps gradually decreases with the number of



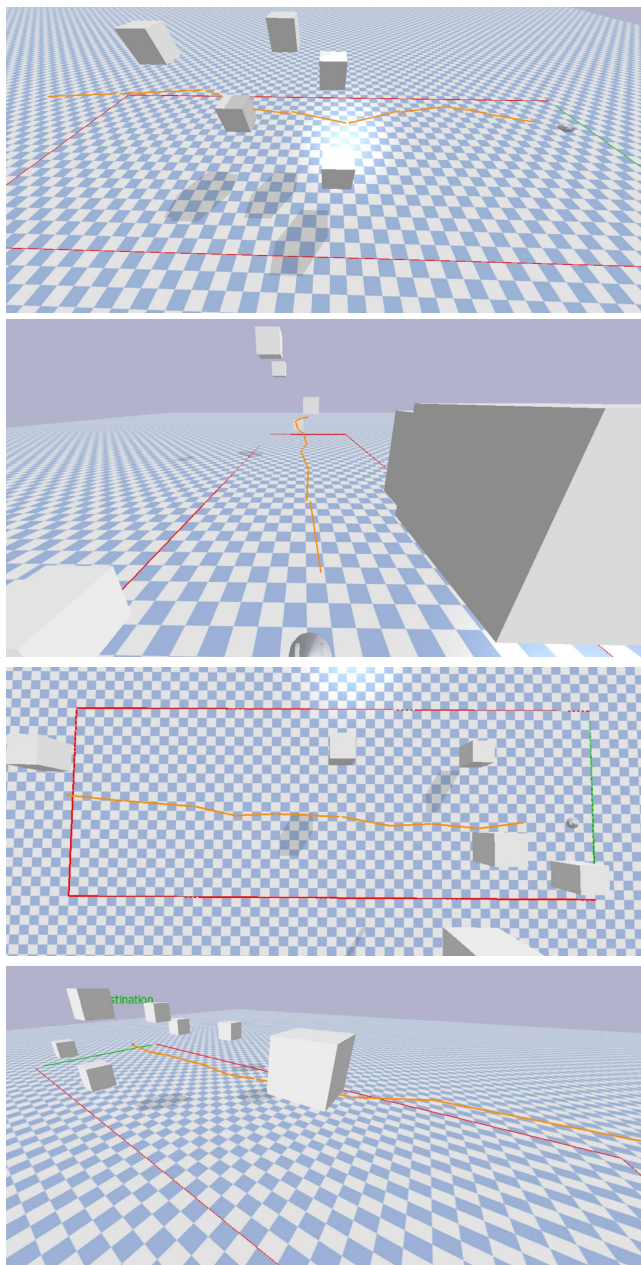
**FIGURE 16.** Average number of steps.

times the target position is reached. After reaching the target position 5,800 times, the average number of steps continued to fluctuate around 490 steps.

Figure 17 is a path-planning diagram for a random dynamic obstacle avoidance scenario. It can be seen that the proposed algorithm still has good passing performance and strong generalization ability in this completely random, complex, unknown environment. This suggests that the algorithm could also be used in a real unknown underwater environment.

In the experimental data for 5,000 runs after training, 2,971 times are successful, and the success rate is 59.42%. The average number of steps is 503. While the success rate is lower than that of the multi-dynamic obstacle scene, the number of steps used is also lower. In this paper, a near-end policy optimization algorithm is used to control the virtual AUV on the map to explore the obstacle avoidance path (instead of directly controlling a real AUV). This approach decouples the obstacle avoidance method from the AUV's propulsion system. The obstacle avoidance method in this paper is applicable as long as the propulsion system can be controlled to follow the path on the map. Regardless of the number of thrusters and the method of propulsion, the obstacle avoidance method greatly improves the algorithm's generalizability.

However, there are inevitably errors in the actual application process. In particular, this paper assumes that AUVs can avoid obstacles in an ideal environment, which means that they can obtain obstacle information without delay and are not affected by dynamic current and other factors in the underwater environment. Therefore, future research can refer to the following latest work. Zhengru Fang et al. formulated a two-stage joint power control, computational resource allocation, and trajectory scheduling for Internet of Underwater Things (IoUT) networks; the approach considered the turbulent ocean environments in the context of a multi-AUV-aided heterogeneous network for energy-efficient information collection [20]. J Wang et al. proposed an active queue management (AQM) policy for the IoUT node in order to reduce the peak age of information (PAoI), beneficially compressing the packets with a long waiting time [21].



**FIGURE 17.** Random dynamic obstacle avoidance scenario path-planning diagram.

G. Han et al. focused on passive attacks in underwater acoustic sensor networks and proposed an autonomous underwater vehicle (AUV)-aided data-importance-based scheme for protecting location privacy (DIS-PLP) [22].

## VI. CONCLUSION

This study builds an obstacle collision prediction model. Based on the results of the simulation of the obstacle inertial motion, the safety of AUV navigation is evaluated to improve the model's sensitivity to dynamic obstacles. The introduction of the long short-term memory network transforms the environmental state into a high-dimensional perception situation, strengthening the network's ability to learn

time-series obstacle avoidance data. Thus, we propose an AUV dynamic obstacle avoidance method based on a CPM-LSTM-PPO algorithm. Using the improved PPO algorithm in an unknown 3D environment with multiple types of obstacles and without any prior knowledge, AUV can find a better path and complete the obstacle avoidance task after repeated trial and error. This is more in line with actual situations and has a high success rate of obstacle avoidance.

## REFERENCES

- [1] H. Lü, J. Xie, J. Xu, Z. Chen, T. Liu, and S. Cai, "Force and torque exerted by internal solitary waves in background parabolic current on cylindrical tendon leg by numerical simulation," *Ocean Eng.*, vol. 114, pp. 250–258, Mar. 2016.
- [2] J. Cai, Y. Zhang, Y. Li, X. Liang, and T. Jiang, "Analyzing the characteristics of soil moisture using GLDAS data: A case study in eastern China," *Appl. Sci.*, vol. 7, no. 6, p. 566, May 2017.
- [3] G. Zhao, Z. Yan, F. Qian, H. Sun, X. Lu, and H. Fan, "Molecular simulation study on the rheological properties of a pH-responsive clean fracturing fluid system," *Fuel*, vol. 253, pp. 677–684, Oct. 2019.
- [4] Z. Qiu, M. Jiao, T. Jiang, and L. Zhou, "Dam structure deformation monitoring by GB-InSAR approach," *IEEE Access*, vol. 8, pp. 123287–123296, 2020.
- [5] L. Ma, H. Zhang, S. Meng, and J. Liu, "Volcanic ash region path planning based on improved A-Star algorithm," *J. Adv. Transp.*, vol. 2022, pp. 1–20, Feb. 2022.
- [6] T. Xu, H. Zhou, S. Tan, Z. Li, X. Ju, and Y. Peng, "Mechanical arm obstacle avoidance path planning based on improved artificial potential field method," *Ind. Robot, Int. J. Robot. Res. Appl.*, vol. 49, no. 2, pp. 271–279, Feb. 2022.
- [7] B. B. K. Ayawli, A. Y. Appiah, I. K. Nti, F. Kyeremeh, and E. I. Ayawli, "Path planning for mobile robots using morphological dilation Voronoi diagram roadmap algorithm," *Sci. Afr.*, vol. 12, Jul. 2021, Art. no. e00745.
- [8] F. Yang, X. Fang, F. Gao, X. Zhou, H. Li, H. Jin, and Y. Song, "Obstacle avoidance path planning for UAV based on improved RRT algorithm," *Discrete Dyn. Nature Soc.*, vol. 2022, pp. 1–9, Jan. 2022.
- [9] D. Agarwal and P. S. Bharti, "Implementing modified swarm intelligence algorithm based on slime moulds for path planning and obstacle avoidance problem in mobile robots," *Appl. Soft Comput.*, vol. 107, Aug. 2021, Art. no. 107372.
- [10] L. Wen, G. Kong, H. Abuel-Naga, Q. Li, and Z. Zhang, "Rectification of tilted transmission tower using micropile underpinning method," *J. Perform. Constructed Facilities*, vol. 34, no. 1, Feb. 2020, Art. no. 04019110.
- [11] P. Sun, K. Zhang, S. Wu, R. Wang, and M. Wan, "An investigation of real-time GPS/GLONASS single-frequency precise point positioning and its atmospheric mitigation strategies," *Meas. Sci. Technol.*, vol. 32, no. 11, 2021, Art. no. 115018.
- [12] W. Yahui, L. Chunyang, and X. Saibao, "Mobile robotic perception and autonomous avoidance based on visual depth learning," *Electron. Meas. Technol.*, vol. 44, no. 20, pp. 99–106, 2021, doi: 10.19651/j.cnki.emt.2107906.
- [13] X. Juntao, L. Zhonghang, C. Shumian, and Z. Zhenhui, "Obstacle avoidance planning of virtual robot picking path based on deep reinforcement learning," *Trans. Chin. Soc. Agricult. Machinery*, vol. 51, no. S2, pp. 1–10, 2020.
- [14] L. Qingjie, L. Youyong, and L. Shaoli, "Research on deep reinforcement learning for intelligent obstacle avoidance scenarios," *Technol. IoT AI*, vol. 1, no. 2, pp. 18–22, 2018.
- [15] S. Lixiang, S. Xiaoxian, L. Chengju, and J. Wen, "Obstacle avoidance algorithm for mobile robot based on deep reinforcement learning in crowd environment," *Inf. Control*, vol. 51, no. 1, pp. 107–118, 2022, doi: 10.13976/j.cnki.xk.2022.0099.
- [16] P. Mirowski, R. Pascanu, and F. Viola, "Learning to navigate in complex environments," in *Proc. Int. Conf. Learn. Represent., Comput. Sci.*, London, U.K., 2017, pp. 1–5.
- [17] Q. Cheng, X. Wang, J. Yang, and L. Shen, "Automated enemy avoidance of unmanned aerial vehicles based on reinforcement learning," *Appl. Sci.*, vol. 9, no. 4, p. 669, Feb. 2019.
- [18] Z. Bin, "Path planning of UAV using guided enhancement Q-learning algorithm," *Acta Aeronautica Astronautica Sinica*, vol. 42, no. 9, pp. 506–513, 2021.

- [19] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
- [20] Z. Fang, J. Wang, J. Du, X. Hou, Y. Ren, and Z. Han, "Stochastic optimization-aided energy-efficient information collection in internet of underwater things networks," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1775–1789, Feb. 2022.
- [21] Z. Fang, J. Wang, C. Jiang, X. Wang, and Y. Ren, "Average peak age of information in underwater information collection with sleep-scheduling," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 10132–10136, Sep. 2022.
- [22] G. Han, Y. Chen, H. Wang, Y. He, and J. Peng, "AUV-aided data importance based scheme for protecting location privacy in smart ocean," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9925–9936, Sep. 2022, doi: 10.1109/TVT.2022.3178379.



**GUOHAO ZHU** received the B.E. degree in software engineering from the Jinling Institute of Technology, Nanjing, China, in 2021. He is currently pursuing the M.E. degree in geomatics and remote sensing engineering with Jiangsu Ocean University, Lianyungang, China.

His research interests include computer vision, image sonar, reinforcement learning, and deep learning.

Mr. Zhu won First Prize in the 7th Jiangsu Provincial College *Journal of Geomatics* Innovation and Entrepreneurship Competition, in 2022.



**ZHOU SHEN** is currently pursuing the bachelor's degree in surveying and mapping engineering with Jiangsu Ocean University, Lianyungang, China.

Her current research interest includes image sonar.

Ms. Shen won the Second in the National English Competition for College Students, the Second in the Cross-Cultural English Competition of Jiangsu Ocean University, and the Third Prize in the Jiangsu Ci Talent Competition.



**LAIYUAN LIU** is currently pursuing the bachelor's degree in surveying and mapping engineering with Jiangsu Ocean University, Lianyungang, China.

His research interests include oceanographic survey and data processing.



**SICONG ZHAO** received the B.E. degree in surveying and mapping engineering from Shijiazhuang Tiedao University, Shijiazhuang, China, in 2021. She is currently pursuing the M.E. degree in geomatics and remote sensing engineering with Jiangsu Ocean University, Lianyungang, China.

Her research interests include mesoscale phenomena in the ocean, mesoscale eddy, and remote sensing image data processing.

Ms. Zhao won the First Prize in the 6th and 7th Jiangsu Provincial College *Journal of Geomatics* Innovation and Entrepreneurship Competition.



**FANGZHENG JI** received the B.E. degree in computer science and technology from the Xuhai College, CUMT, Xuzhou, China, in 2020. He is currently pursuing the M.E. degree in geomatics and remote sensing engineering with Jiangsu Ocean University, Lianyungang, China.

His research interests include machine learning, geographic information systems, and intelligent systems.

Mr. Ji won first in the 7th Jiangsu Provincial College *Journal of Geomatics* Innovation and Entrepreneurship Competition, in 2022.



**ZIXIA JU** is currently pursuing the bachelor's degree in surveying and mapping engineering with the School of Marine Technology and Geomatics, Jiangsu Ocean University. She won the First Prize in the 12th National College Students' Surveying and Mapping Science Paper Competition of "South Surveying and Mapping Cup" and the 2022 National College Students' Surveying and Mapping Science Innovation and Entrepreneurship Intelligence Competition

Development and Design Competition.



**JIALONG SUN** received the Ph.D. degree in science and technology of surveying and mapping from the Shandong University of Science and Technology, Qingdao, China.

He is currently the Dean of the School of Marine Technology and Geomatics, Jiangsu Ocean University, Lianyungang, China. He has published more than 30 academic articles in the *Journal of Coastal Research, Territorial, Atmospheric and Oceanic Sciences*, and other domestic and foreign

journals. He published two academic monographs, one monograph on educational reform and applied for 15 Chinese invention patents. His research interests include marine geodesy, marine intelligent surveying, and mapping equipment.

Dr. Sun has won more than 20 awards and honorary titles, including the Second Prize of National Surveying and Mapping Science and Technology Progress Award, the Second Prize of Marine Science and Technology Award, the First National University GIS Teaching Achievement Grand Prize, the Second Prize of Satellite Navigation and Positioning Teaching Achievement, the Second Prize of National Surveying and Mapping Discipline Young Teachers Teaching Contest, and the "Excellent Teacher" of Jiangsu Ocean University.

...